Temporal Tracking With Implicit Templates

Hussein Yahia, Isabelle Herlin, Laurent Vogel

INRIA
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Temporal Tracking With Implicit Templates.

H. M. YAHIA, I. L. HERLIN, L. VOGEL

N° 2701

25 Octobre 1995

PROGRAMME 4

ISSN 0249-6399

ROCQUENCOURT

Temporal Tracking With Implicit Templates.

H. M. YAHIA, I. L. HERLIN, L. VOGEL

Programme 4 — Robotique, image et vision
Projet AIR

Rapport de recherche n° 2701 — 25 Octobre 1995 — 27 pages

Abstract: This work is devoted to the segmentation and tracking of structures undergoing large deformations in an image sequence. The overall goal is the modelling of complex moving and deforming structures that simultaneously brings compression of the data, segmentation and tracking of the structures on temporal sequences of images. Deforming structures are approximated by implicit templates, defined as level sets of some particular implicit functions. Implicit functions offer an interesting tool for the modelling of visco-elastic structures. Structures are approximated using an energy minimization process; the energy functional carries information about position, tangent and curvature. The implicit functions used in this work are defined by some control points, their radius of influence, and an iso-value. Once the structures are segmented, tracking is modelled using the displacements of the control points and the variation of the radii of influence. The method is first applied on echocardiographic images, providing promising results for segmentation. We also apply the method on a sequence of images displaying a vortex formation, and we show that such a vortex structure can also be segmented and tracked with implicit functions.

Key-words: Segmentation, Implicit Function, Temporal Tracking, Ultrasound Image, Remote Sensing, Vorticity, Energy Minimization, Conjugate Gradient Method

(Résumé : tsvp)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)

Suivi Temporel Par Contours Implicites.

Résumé : Nous nous intéressons à la segmentation et au suivi temporel de structures déformables dans une séquence d'images. Les structures sont approximées par des contours implicites, qui permettent une modélisation des déformations visco-élastiques et des changements de topologie. Les contours implicites sont générés par la minimisation d'une fonctionnelle d'énergie. L'énergie utilisée ici tient compte des informations de position, tangente et courbure. Les contours implicites sont définis par des points de contrôle et des rayons d'influence. Le problème du suivi temporel des structures est alors ramené à celui de la variation des paramètres géométriques des points de contrôle. Cette méthode est appliquée sur des images d'échographie, puis sur une suite d'images montrant la formation d'un tourbillon. Ce travail constitue la première étape vers une modélisation des structures complexes offrant simultanément compression des données, segmentation et suivi des structures dans des séquences temporelles d'images.

Mots-clé : Segmentation, Fonctions Implicites, Suivi Temporel, Images Echographiques, Imagerie Satellitaire, Tourbillon, Minimisation d'Energie, Gradient Conjugué

1 Introduction

In this paper we address the problem of segmenting deformable moving structures in an image sequence: we are looking for a mathematical framework allowing simultaneously segmentation, global tracking and data compression. Moreover, we are interested in deforming structures displaying small and large deformations. For this purpose we study the approximation of structures by implicit templates. Recent research in this promising field used parametric modelling (like snakes and spline curves for instance) or probabilistic models (like Markov random fields) [7, 16, 11]. The use of these models resulted in interesting image representations for segmentation, but the ability of using such models for incorporating knowledge about motion is strongly limited by the very specific geometric limitations of parametrized curves and surfaces. Moreover we believe that implicit functions offer more potential capabilities for animating visco-elastic deformations in Image Processing. To give an example, it is not easy to relate an elastic or visco-elastic behavior of a spline curve to the motion of its control points and tangents. In a same way, snake template parameters can be used in the segmentation process of a still image, but it would be difficult to relate the motion of an image structure to the variation of the parameters of the snake, specially when topological changes appear or if the structure undergoes large deformations. Of course implicit objects also possess local parametrizations (and, in the case of 1-dimensional implicit objects, global parametrizations also exist and are related to the topological notion of connectedness), but, in a sense, implicit templates are "orthogonal" to parametrized curves and surfaces: their modelling capabilities do not rely on how a given function maps and "distorts" a parameter interval, but rather on the geometry of the graph of a higher-dimensional function whose level sets define the topology and geometry of the implicit shape. The geometry of the graph of the higher-dimensional function often encodes in a simple way complicated topological changes in each of its sections. This is the reason why implicit functions are so often used in Computer Graphics to model complex topological changes, whereas any attempt in attaining such capabilities with parametrized curves and surfaces would be fraught and painful. We note that some recent work [20] used implicit functions to model growing bulbs for image segmentation. Here we address the problem from a different point of view, by using implicit functions defined by a small number of control points (in the view of data compression). The effectiveness of implicit functions to handle complex topological changes and visco-elastic geometric simulation is the reason why we toyed with the possiblity of using them in Image Processing for the segmentation and tracking of deformable structures.

Implicit functions have been a subject of intense research, particularly in the domain of Computer Graphics. Apart the geometrical capabilities we have already mentionned, they also easely implement membership algorithms. They tend to become a standard modelling tool in Computer Graphics, where, like splines curves and surfaces, they display the interesting property of designing complex shapes with a minimal amount of encoding data. In Image Processing, the cost-effectiveness of implicit functions can be used for compression, and their potential richness for designing complex and moving shapes foreshadows the use of implicit functions for the segmentation of moving structures. The effectiveness of implicit functions to achieve modelling of deforming and rubber-like objects is the reason why some classes of implicit functions are called Soft Objects [22, 23, 24]. They easily handle multiple connected components, and can be generated at reasonnable cost.

In this paper we study the problem of approximating an object (a cardiac cavity on an ultrasound image, a vortex formation on a sequence of images), or a set of pixels (the list of edge points obtained by an edge detector), by implicit functions. The paper is organized as follows: in section 2 a basic review of the properties of implicit functions is presented, and section 3 develops some aspects about implicit contours that are of particular importance for our subject. In section 4, the approximation problem is described in terms of an energy minimization process, and results on synthetic contours are displayed. Section 5 is devoted to the approximation technique on edge points over a sequence of echocardiographic images. And in section 6 the model is used to perform tracking of a moving vortex structure. The last part is devoted to discussion and conclusion.

2 Review of Implicit Functions

Since this paper is concerned with 2D templates, we only discuss 2D implicit contours.
Let $U \subset I\!R^2$ be an open set and:

$$\phi : U \longrightarrow I\!R$$

be a real valued function.
An iso-contour associated to ϕ is a set of points $M \in U$ such that $\phi(M) = c$ where c is a quantity called iso-value. Thus an iso-contour is simply a level set of ϕ. Usually, ϕ is choosen C^∞. Note that the use of C^∞ functions does not put restrictions on the resulting iso-contours [2]: implicit templates are often defined by local functions incorporating a given metric. By choosing a L^p metric with p high, it is even possible to obtain narrow corners with implicit

functions.

In the Computer Graphics literature, several authors use different kind of functions:

- super-ellipsoids [22]:

$$\phi(x,y) = \frac{x^k}{a^k} + \frac{y^k}{b^k} \tag{1}$$

- super-quadrics [1], defined in 3D by:

$$\phi(x,y,z) = \left[x^{\frac{2}{m}} + y^{\frac{2}{m}}\right]^{\frac{m}{n}} + z^{\frac{2}{n}} \tag{2}$$

However, the use of a global analytic expression for ϕ looses some control over the resulting shapes. To make implicit contours more tractable in modelling, different authors proposed implicit functions defined more locally. Basically, one can distinguish two broad classes of implicit functions defined in this way: those defined with *control points* and those defined with *skeletons* [9]. Since the minimization process we have in mind is more easily described with control points, we only focus on that latter formulation.

Let S be a finite set of points in U, called control points. The implicit function is then defined by an expression like:

$$\phi(M) = \sum_{k_i \in S} \phi_i(M) \ ,$$

where ϕ_i is a local function defined in a neighborhood of k_i. Different authors have proposed variations on this formulation:

- Blinn [3] uses a formulation where ϕ_i is defined by an exponential. The resulting objects are called Blobby Models,
- Nishimura and al. [15] use a polynomial expression for ϕ_i. The implicit contours obtained this way are often called Metaballs,
- Wyvill and al. [24] use another polynomial formulation. The resulting objects are called Soft Objects. This is the formulation we are going to use in this paper. We describe it in more detail in the next section.

We also note that Cohen and Cohen [6] use another implicit formulation which does not fit into the categories mentionned above: the authors use a global formulation of the implicit function and generate intermediate points to control the shape of the implicit object. Also, Sclaroff and Pentland [18] introduced generalized implicit functions, and an efficient technique is described to apply various transformations on implicit objects.

In the next section, we describe in greater detail the implicit functions called Soft Objects and develop some aspects of implicit functions that will be used in the energy formulation.

3 Soft Objects

Blobby Models, Metaballs and Soft Objects are very similar formulations. In this paper we choose Soft Objects, defined by local functions which vanish outside a bounded region. As mentionned before, we consider a finite set:

$$S = \{k_1, k_2, \cdots, k_n\}$$

of control points in U. To each control point k_i we associate a *radius of influence* r_i. Each local contribution ϕ_i is defined by:

$$\phi_i(M) = a\frac{\|\overrightarrow{k_iM}\|^6}{{r_i}^6} + b\frac{\|\overrightarrow{k_iM}\|^4}{{r_i}^4} + c\frac{\|\overrightarrow{k_iM}\|^2}{{r_i}^2} + d \ , \tag{3}$$

if M is such that $\|\overrightarrow{k_iM}\| \leq r_i$, and 0 otherwise. Coefficients a, b, c and d are chosen so that the function of one real variable:

$$f(x) = ax^3 + bx^2 + cx + d$$

is C^1 and looks like the function depicted in figure 1. The determination of a, b, c and d involves the resolution of a linear system. One finds [24] $a = -0.44$, $b = 1.88$, $c = -2.44$ and $d = 1$. The use of squared terms of the distance, instead of the distance itself, facilitates

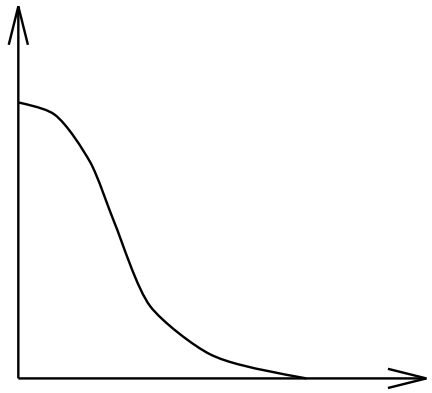

Figure 1: *Graph of function f .*

the computations of various parameters associated with the function such as its derivatives, while it does not change the qualitative properties of the function.
Function ϕ is then defined by:

$$\phi(M) = \sum_{k_i \in S} \phi_i(M) \tag{4}$$

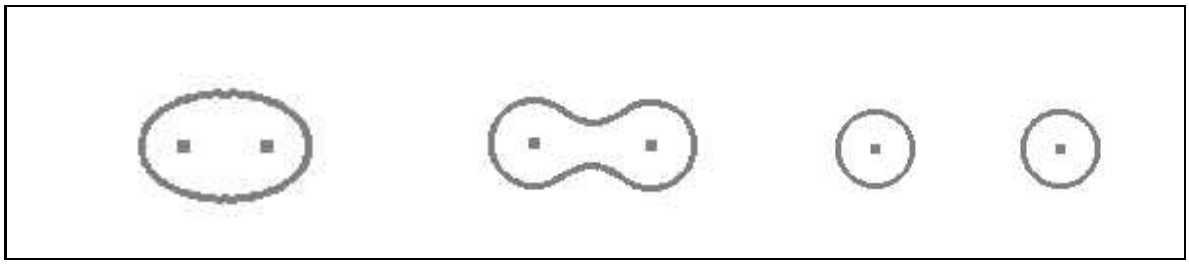

Figure 2: *Iso-contours obtained with two positive control points: given an iso-value, two positive control points and their radius of influence, the shape of the iso-contour changes continuously and finally can break itself into two connected components if the control points are too far away from each other.*

If we interpret ϕ as a physical potential field, the forces that correspond to this potential are never repelling, since each contribution ϕ_i is positive. Hence each control point acts as an attractive center force field of bounded support. To incorporate repulsive forces, which can be quite useful for controlling concave parts of an iso-contour, we introduce **negative** control points: these are just ordinary control points, but their associated field function ϕ_i is negative. To put it in another way, we eventually define function ϕ by:

$$\phi(M) = \sum_{k_i \in \mathcal{S}^+} \phi_i(M) - \sum_{k_i \in \mathcal{S}^-} \phi_i(M) \ ,$$

where $\mathcal{S}^+$ and $\mathcal{S}^-$ are respectively the set of positive and negative control points ($\mathcal{S} = \mathcal{S}^+ \cup \mathcal{S}^-$). Negative control points are introduced because they reduce the number of control points needed to model concave parts of an object: without negative control points, such regions are obtained at the expense of adding much more positive control points, thus increasing the amount of encoding data.

If S contains only one point, the non-empty iso-contours are simple closed contours. With two control points, one obtains different types of iso-contours, depending on the location of control points and the radius of influence's values. See figure 2.

We now focus on differential attributes of iso-contours. Firstly, let us conform to standard usages in differential geometry [21]: if $p \in U$, a vector based at p is a pair $(p, \vec{v})$ where $\vec{v}$ is a vector in $\mathbb{R}^2$. Hence a vector based at p is simply a vector whose tail is at p rather than at the origin.

To every function

$$\phi : U \longrightarrow I\!R$$

is associated the gradient vector field in U: at each point $p \in U$ we associate the gradient of ϕ based at p, that is to say the vector

$$(p, \nabla\phi(p)) = (p, \frac{\partial\phi}{\partial x}(p), \frac{\partial\phi}{\partial y}(p)) \tag{5}$$

Such a vector based at p is always orthogonal to an iso-contour passing through p and gives the direction of steepest variation of ϕ at p. Thus we see that, to get a tangent vector of an iso-contour at a given point p, one can simply compute the gradient at p and rotate it around p by an amount of $-\frac{\pi}{2}$.

In the minimization process described in the following section, we also make use of the curvature information of an iso-contour. Such a quantity can be computed at each point using the second fundamental form. Let C be an iso-contour of an implicit function ϕ. An orientation of C is given by the choice of a normal vector, for instance $\frac{\nabla\phi}{\|\nabla\phi\|}$. With this orientation, the second fundamental form of C at p on a tangent vector $\vec{v} = (p, v_1, v_2)$ is [21]:

$$\mathcal{S}_p(\vec{v}) = -\frac{1}{\|\nabla\phi(p)\|}\left[\ \frac{\partial^2\phi}{\partial x^2}(p){v_1}^2 + 2\frac{\partial^2\phi}{\partial x\partial y}(p)v_1v_2 + \frac{\partial^2\phi}{\partial y^2}(p){v_2}^2\ \right] \tag{6}$$

and the curvature at p is:

$$\frac{\mathcal{S}_p(\vec{v})}{\|\vec{v}\|^2}\ . \tag{7}$$

We note that in [20] a different formula is used to compute the curvature: they merely use the fact that, for 1-dimensional implicit contours, curvature and mean curvature are the same, and compute accordingly the curvature by taking the divergence of the normal vector field. Obviously, in the case of 1-dimensional implicit contours, the two formula are equivalent, but our presentation generalizes to higher dimensions as well. These computations are valid as long as the gradient vector does not vanish at a given point of an iso-contour. In studying differential attributes of an iso-contour, we tacitely make the assumption that the iso-contour is made of regular points. This is justified by observing that singular points correspond to exceptional geometric features we'll hardly encounter in our images. At a singular point of an iso-contour, that is to say a point where the gradient is zero, there is no well defined tangent space and curvature is not defined.

Note also that the membership problem is trivially solved for an iso-contour C: suppose we want to know wether a test point M is inside or outside an iso-contour $C = \phi^{-1}(c)$,

where c is the iso-value associated to C. One just computes the value of ϕ at the test point M, and the decision is made depending on the sign of $(\phi(M) - c)$. This property of implicit objects make them particularly well suited for surfacic representations in 2D space and volumic representations of objects in 3D space, such as those used in CSG (Constructive Solid Geometry) systems.
Several techniques have been proposed to draw an implicit object by computer [4, 12, 24]. In this paper we use the algorithm described in Wyvill and al. [24]: the image is first subdivided into a regular grid of small squares. The value of $(\phi(M) - c)$ is computed at each vertex of each square. We record only those squares for which there is a change of sign of this quantity at some vertex. Then, for each square recorded, an edge is generated which is a linear approximation between vertices inside the square of the iso-contour.

In the next section, we present the energy formulation of the approximation problem by implicit functions.

4 An Approximation Problem

Suppose we are given a polygonal contour Γ in the plane, defined by its set of vertices:

$$V = \{P_1, P_2, \cdots, P_m\}$$

We are going to solve the following problem:

$$\textit{Find a good approximation of } \Gamma \textit{ as an iso-contour } \phi^{-1}(c) \ .$$

By "good approximation", we mean an iso-contour whose shape looks like that of Γ. We seek a function ϕ in the form described in the previous section: a set S of control points (positive and negative), their associated radius of influence $(r_i)_{1 \leq i \leq n}$ and an isovalue c.
As explained before, we are going to express this approximation problem as an energy minimization problem. The energy will be the sum of three terms, described in the following subsection.

4.1 Influence of Position, Tangent and Curvature

Since the iso-contour must be very close to each point of Γ, each value $\phi(P_i)$ must be close to c. Hence we introduce a first term in the energy (see [14]):

$$E_1 = \sum_{i=1}^{m} (c - \phi(P_i))^2 \tag{8}$$

Energy E_1 constrains the value of ϕ at the vertices of Γ only. That can be quite satisfactory for still images, as we'll see in the next subsection, but to accurately track moving structures, it is often necessary to put constraints on tangent and curvature information.

Let us discuss the tangent first. Since Γ is polygonal we can either use the vector $\overrightarrow{P_iP_{i+1}}$ or the tangent of a C^1 curve interpolating Γ as the tangent direction (denoted $\overrightarrow{T_i^1}$) at P_i on the original contour Γ. We use the first solution, since it gives very good results and is cheaper to implement. On the iso-contour we recall that tangent vectors (denoted $\overrightarrow{T_i^2}$) are given by the gradient rotated clockwise by a right angle. The energy functional tries to keep these tangent vectors $\overrightarrow{T_i^1}$ and $\overrightarrow{T_i^2}$ in the same direction. We introduce a second term in the energy:

$$E_2 = \sum_{i=1}^{m-1} [\ det(\overrightarrow{T_i^1}, \overrightarrow{T_i^2})\ ]^2 = \sum_{i=1}^{m-1} [\ det(R_{-\frac{\pi}{2}}(\nabla\phi)(P_i), \overrightarrow{P_iP_{i+1}})\ ]^2 = \sum_{i=1}^{m-1} \begin{vmatrix} \frac{\partial\phi}{\partial y}(P_i) & t_x^i \\ -\frac{\partial\phi}{\partial x}(P_i) & t_y^i \end{vmatrix}^2 \quad (9)$$

where $R_{-\frac{\pi}{2}}$ is the right angle rotation, and $(t_x^i,\ t_y^i)$ are the components of $\overrightarrow{P_iP_{i+1}}$.

Now for the curvature. Since curvature involves second derivatives, it is necessary to approximate the polygonal contour Γ by a C^2 curve in order to compute curvature on the original contour Γ. We use a simple cubic B-spline interpolation scheme [5], which defines an interpolating curve γ by a set of curves γ_i which can be expressed in a matrix form:

$$\gamma_i(t) = [\ t^3\ t^2\ t\ 1\ ] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix}$$

It should be noted, however, that there exists other methods to approximate the curvature of a polygonal contour. For instance Maillot, Yahia and Verroust [13] make use of a least square method to approximate the matrix of the second fundamental form at each point. Nevertheless we found cubic interpolation B-splines quite satisfactory for synthetic contours, because such contours are "clean enough" to undergo direct interpolation. Noise present in natural images eliminates any attempt of using interpolation in a clean stroke: such an interpolating contour would display enormous variations in tangent and curvature. Instead, an approximation scheme must be used, as one expects a regular "variation diminishing" spline to compute the curvature energy. We will resume this discussion in section 5. Now, let γ be the parametrized cubic B-spline approximating (or interpolating, depending on a "synthetic" or "natural image" context) Γ and let $(t_i)_{1\leq i\leq q}$ be a sequence of parameter values.

The curvature C_i^1 of the parametrized curve γ at $p = \gamma(t_i)$ is

$$\frac{< \ddot{\gamma}(t_i) \mid \overrightarrow{N(\gamma(t_i))} >}{\| \frac{d\gamma}{dt}(t_i) \|^2} ,$$

$\overrightarrow{N(\gamma(t_i))}$ being the unit normal vector at $\gamma(t_i)$, and $< \cdot \mid \cdot >$ denoting the dot product of vectors. Since the curvature C_i^2 of the iso-contour is obtained from the second fondamental form (see section 3), we introduce a third energy term:

$$E_3 = \sum_{i=1}^{q} (C_i^1 - C_i^2)^2 = \sum_{i=1}^{q} \left[\frac{< \ddot{\gamma}(t_i) \mid \overrightarrow{N(\gamma(t_i))} >}{\| \frac{d\gamma}{dt}(t_i) \|^2} - \frac{\mathcal{S}_p(\vec{v})}{\|\vec{v}\|^2} \right]^2 \tag{10}$$

where $\vec{v}$ is a tangent vector at the iso-contour passing through $p = \gamma(t_i)$, and $\mathcal{S}_p$ the second fundamental form of the iso-contour. If energy E_3 is minimized, then at each point $p = \gamma(t_i)$, the curvature of the iso-contour will be close to that of the parametrized spline.

The final form of energy we use is

$$E = \alpha E_1 + \beta E_2 + \gamma E_3 \tag{11}$$

with $\alpha + \beta + \gamma = 1$.

We minimize E on the following set of variables: the coordinates (x_i, y_i) of the control points k_i and their radii of influence r_i. Energy E is then a function of all these variables:

$$E(x_1, y_1, x_2, y_2, \cdots, x_n, y_n, r_1, r_2, \cdots, r_n) .$$

Since E is a positive definite sum of squares, whose partial derivatives

$$\frac{\partial E}{\partial x_i} , \frac{\partial E}{\partial y_i} , \frac{\partial E}{\partial r_i}$$

are easily computed (see Appendix), we use a conjugate gradient method to minimize E (see [10]). After having done some experiments, we found that the initialization of the control points and of their radii of influence plays a crucial role in the quality of the result. A good starting initialization can be generated in the following way. Positive control points are placed inside the convex parts of Γ, and negative control points outside the concave parts of Γ. See figure 3. For each control point, either positive or negative, we compute the distance d between that control point and Γ. Then the radius of influence of that control point is $d+K$, where K is a fixed constant. (Typically, K is choosen in order to insure that Γ lies inside the circles defined by the control points and their radius of influence). In this paper, we use

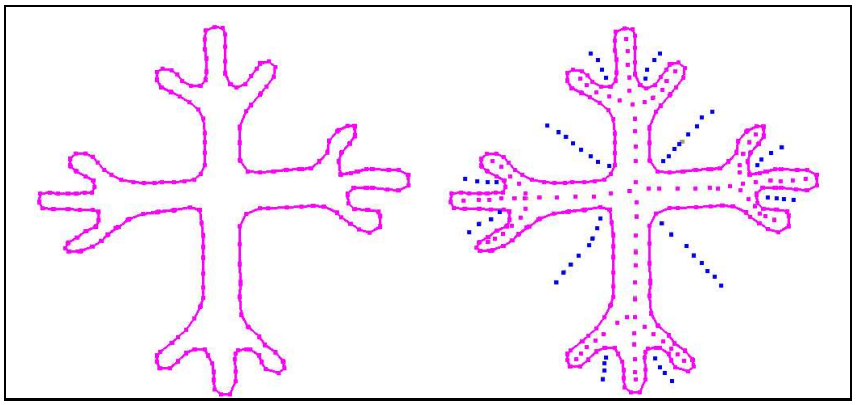

Figure 3: *Left: a contour (Merovingian triceros). Right: initial arrangement of control points. Blue (negative) control points are located outside in the concave parts of the contour, while positive control points are inside the contour.*

a value of $K = 15$ pixels. Also, in the examples shown, we keep a fixed iso-value. For the particular type of initialization just described, we found that an iso-value $c = 0.9$ gives very good results for a wide variety of contours (Recall that the maximum value of each $\phi_i = 1$).

After having done various experiments, we found that taking $\alpha = 0.7$, $\beta = 0.2$ and $\gamma = 0.1$ gives quite satisfactory results, although one could obviously find some examples where better results are obtained by changing these tuning coefficients.

4.2 Discussion of the method on synthetic contours

As explained before, energy E is minimized on the following set of variables:

$$(x_1, y_1, x_2, y_2, \cdots, x_n, y_n, r_1, r_2, \cdots, r_n)$$

We found that better results are obtained by breaking the minimization process down to two steps corresponding to the minimization on the (x_i, y_i) first, then on the r_i. Hence the minimization loop takes the following form:

```
while ( error > THRESHOLD ) {
   error = conjugate_gradient_on_the_positions_only;
   error = conjugate_gradient_on_the_radii_only;
}
```

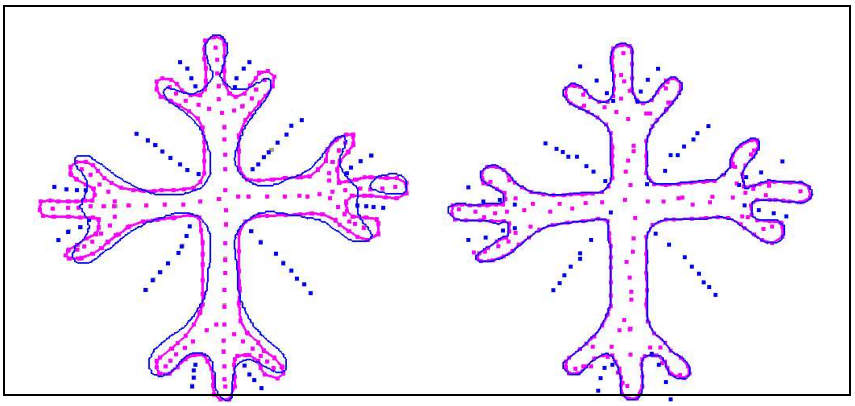

Figure 4: *Result of the minimization process with energy E_1 and the Merovingian triceros displayed at the left in figure 3. Left: initialization of control points. The blue template is the implicit template corresponding to the initialization. Here the iso-value $c = 0.9$. Right: Result of the minimization process with energy E_1.*

For still images, energy E_1 is often quite satisfactory, as shown in figure 4. But for temporal tracking on moving structures, energy E_1 does not put enough constraints on the control points' motion. These are free to move about in some various directions without taking care of the location of neighbouring control points. We illustrate this phenomenon in figure 5.[1] To generate these images, we took three consecutive pictures of a vortex formation and segmented them with energy E_1 (this process will be explained in another section). In figure 5, except for the first image, the result of the minimization process of an image is given as an initialization for the minimization of next picture. Since E_1 it is underconstrained, it does not well represent any kind of internal energy of the motion, since such an internal energy would clearly superimpose geometric constraints on tangency and curvature. As a result, the motion of the control points does not carry any information about the physical behaviour of the structure (some control points drift too far outside the convex parts of the structure, while some other control points display large displacements though some of their close neighbours remain fixed). Also, during the motion, energy E_1 does not take care very well about the number of connected components of the resulting shapes. Indeed, from the Gauss-Bonnet theorem [21], we know that, in the case of 1-dimensional implicit objects,

[1] In figures 5, 6, 15 and 16 only the most significant part of the iso-contour is displayed. Indeed, the iso-contour is a closed curve, but we do not draw the part of that iso-contour that stretches to the left of positive (red control points). This is done by a straigthfoward modification of the iso-contour drawing algorithm. In other words, we only display the part of the iso-contour that lies in-between red and green control points.

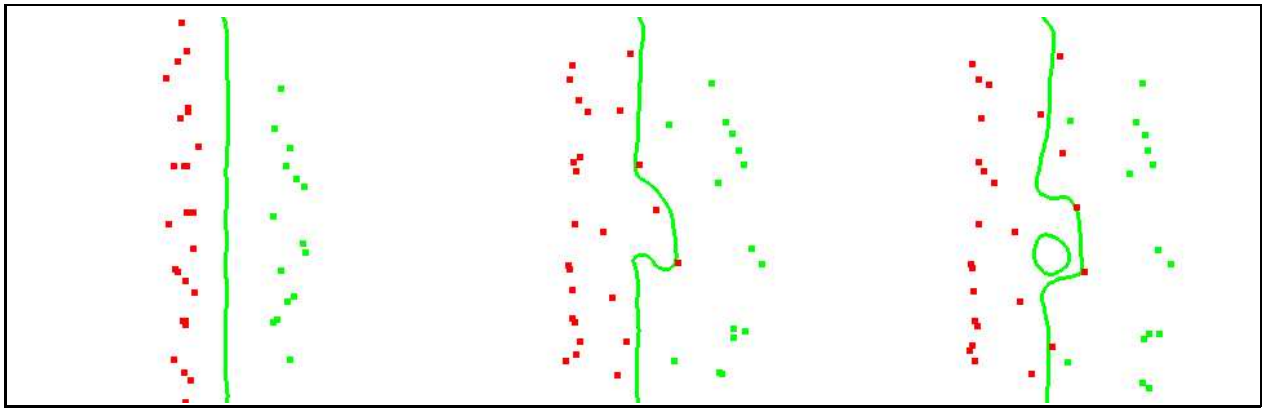

Figure 5: *Energy E_1 doesn't control properly the motion of control points. Positives control points are depicted in red, negative in green. The final iso-contour is made of two connected components.*

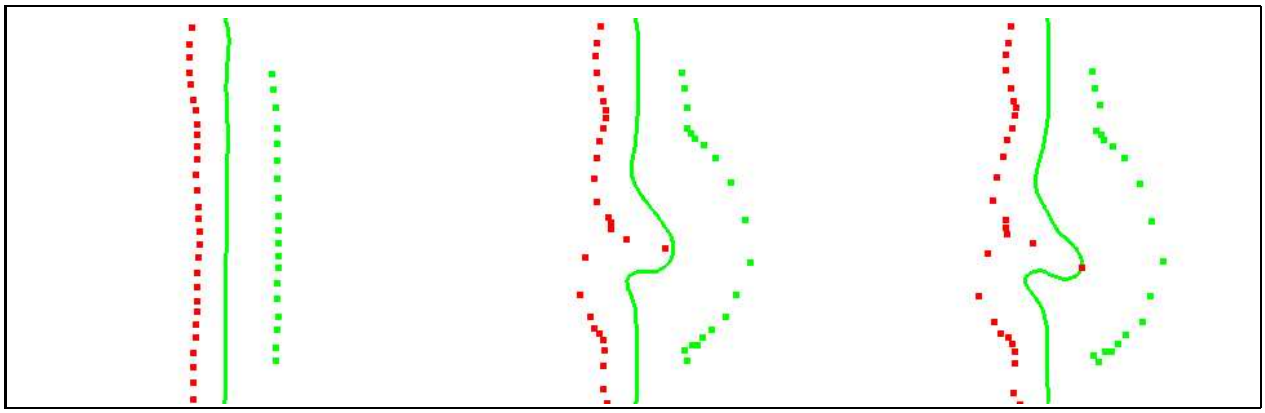

Figure 6: *Energy $E = 0.7E_1 + 0.2E_2 + 0.1E_3$ puts much more constraints on control points displacements.*

the number of connected components of an implicit contour is related to the integral of the curvature, a quantity not taken into account by E_1. This is the reason why the iso-contour is eventually made of several connected components in the rightmost image of figure 5. If we now resume the segmentation of the vortex creation but with energy E instead (figure 6), we see that control points that influence some region of the structure cannot undergo too large random displacements. We make the conclusion that energy E is a better candidate for segmenting moving structures.

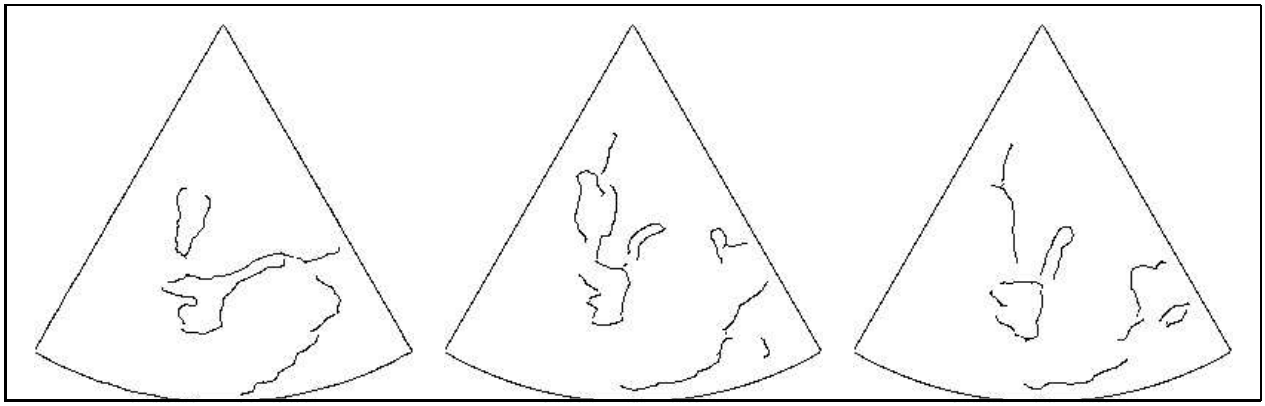

Figure 7: *Result of gradient filtering and hysteresis-thresholding on ultrasound images.*

5 Segmentation of Ultrasound Images

We now apply the method described in previous sections on ultrasound images. Different authors have already studied the problem of segmenting cardiac structures in ultrasound image sequences [11, 19]. For instance in [11] a stochastic model is presented to segment and track specific anatomical structures in temporal sequences of echochardiographic images. Here we start from a sequence of echocardiographic images acquired during the cardiac cycle. These images were provided by Henri Mondor Hospital in Paris. We use a classical Canny-Deriche [8] filter to compute the extrema of gradient norm values. Then, an hysteresis thresholding is applied to get the edges displayed in figures 7 and 8. The contour we called Γ in the previous sections is now made of a collection of contours $(\Gamma_i)_{1\leq i\leq r}$. At this point, it is important to note that the C^2 interpolating curve used in section 4 for synthetic contours is not adequate in the case of range data images. Indeed, due to the noise present in natural images, such an interpolating curve can show enormous variations in curvature, so that during the process of minimizing E, the contribution of E_1 and E_2 become meaningless. Instead, one must use an approximation spline scheme [17]. The result of gradient filtering and hysteresis-thresholding may produce undesirable contour points however located within the support of the implicit function defined by the initialization. These contrived contour points are often just around the structure of interest. To diminish the influence of such contour points, one can merely give more importance to the contour points that are closer to the initialization, since we don't want the resulting iso-contour going to far away from the initialization of the cardiac cavity. This is done by multiplying each contribution of such contours in the energy function

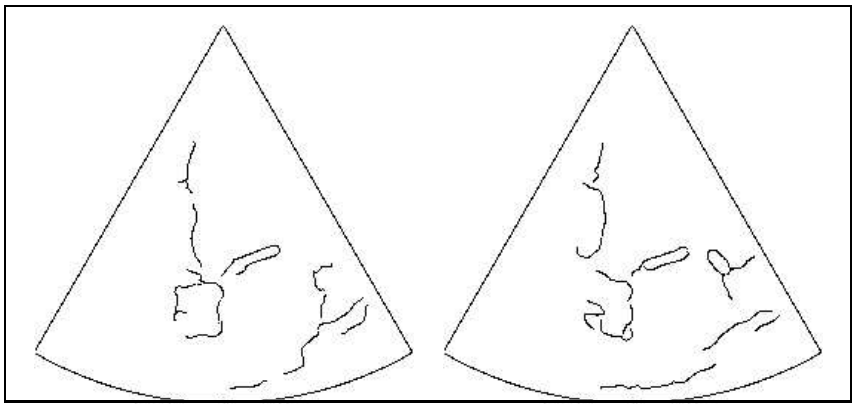

Figure 8: *Result of gradient filtering and hysteresis-thresholding on ultrasound images.*

by a quantity which decreases as the inverse distance of the contour to a point located at the center of the initialization. Note that we also normalize energy E by dividing by the total number of points of Γ.

Figure 9 shows the initialization of control points. In this particular example the control points are chosen by the user. The initial iso-value is choosen so that the resulting iso-contour lies completely inside the cardiac cavity. In this application, we only use positive control points, because the introduction of negative control points does not give significantly better results. Figures 10 and 11 show the result of the minimization on the image sequence. Note that in this example only five control points are used. The cavity is adequately represented by an implicit function at each stage of the deformation, but we note that in this application, the structure doesn't display large deformations. In the next section, we study another example where a structure undergoes more important deformations.

6 Segmentation of a moving vortex

We are now concerned with vortex segmentation on remote sensed images. A vortex isn't convex, and as time proceeds, the structure can break itself into several connected components. Let us consider the image sequence shown in figure 12 (courtesy European Optical Society). These images are synthetic pictures of a vortex evolution. To generate these images,

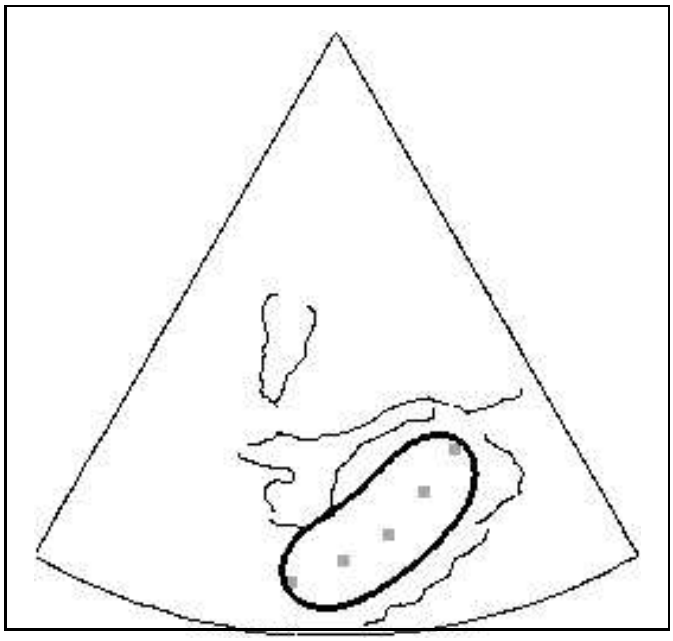

Figure 9: *Initialization of the implicit contour inside the cardiac cavity.*

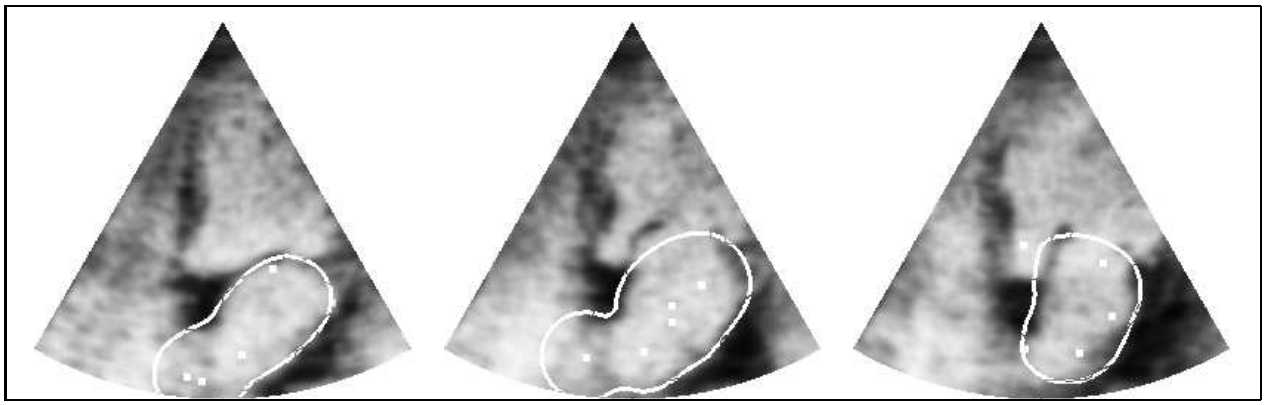

Figure 10: *Result of the minimization process.*

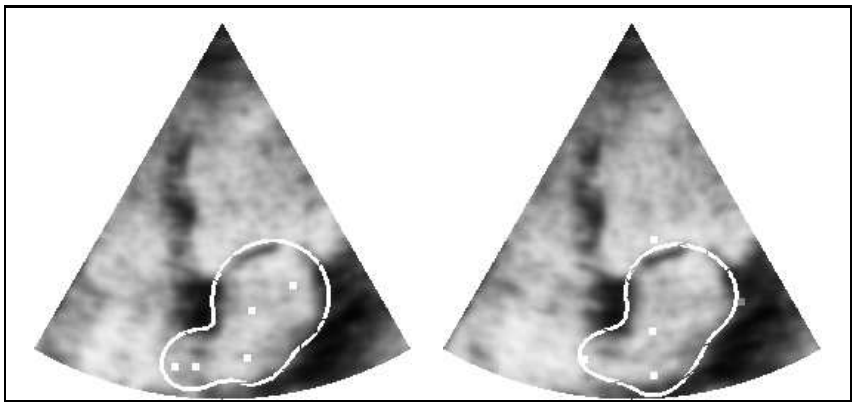

Figure 11: *Result of the minimization process.*

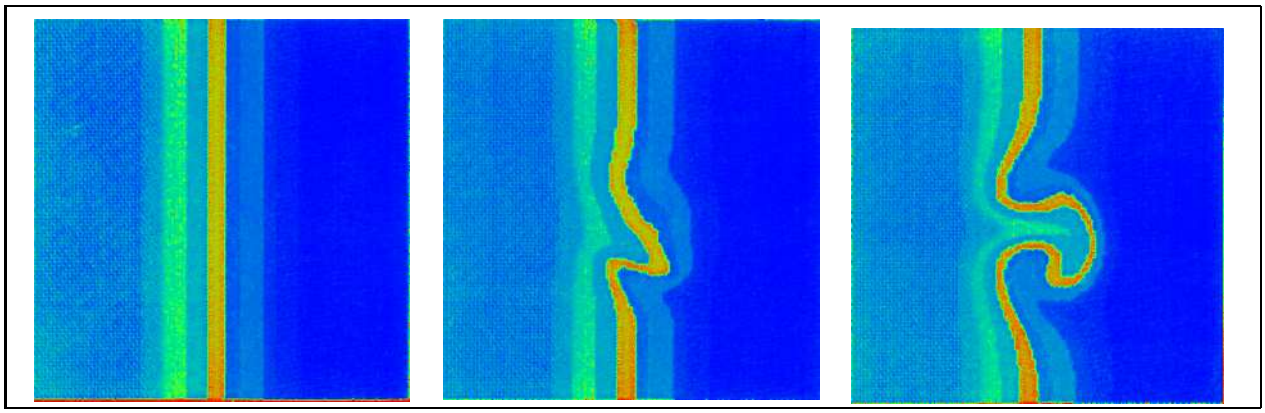

Figure 12: *Synthetic vortex evolution.*

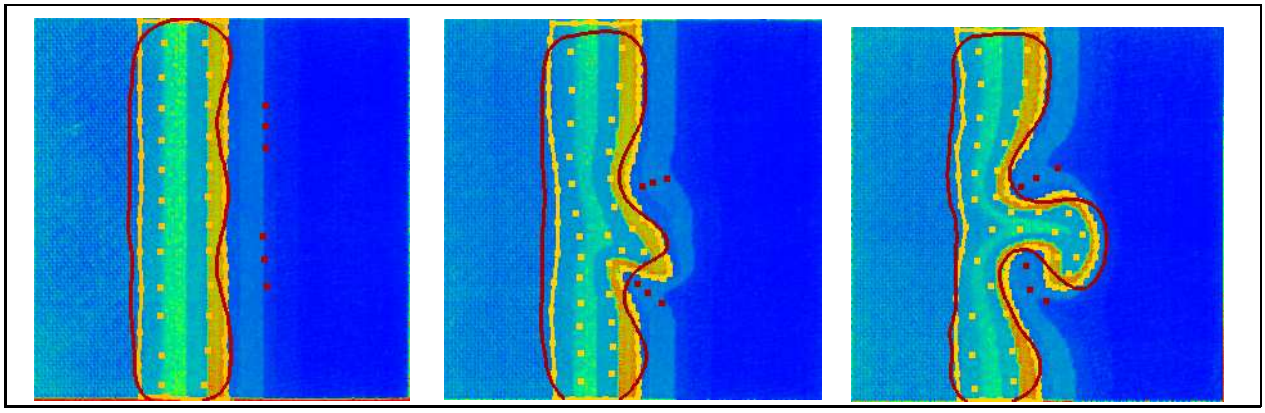

Figure 13: *Initialization of positive and negative control points, and the resulting iso-contour.*

a physical simulation was used to compute the vortex evolution that shows up out of a "cold-hot" polarity. It is only a simulation, but with the advantage of reducing the noise present in natural images. To segment such pictures, we proceed in the same manner as we did for ultrasound images: we compute the edges, then sets of pixels are extracted and approximated with a spline function. Firstly, in figures 13 and 14, we demonstrate the effectiveness of implicit functions to segment a vortex structure. Initializations (made by the user) are shown in figure 13, and the result of the minimization process is displayed in figure 14. Negative control points play an important role in the modelling of the concave parts of the vortex. Secondly, in figures 15 and 16, we demonstrate the ability of implicit functions to track the motion of a deforming structure. To generate these images, we first segmented the leftward picture in figure 12, then the result of that minimization process is given as the initialization step for the middle picture, which is then segmented. The result of that latter segmentation is again given as an initialization for the rightmost picture which is also segmented. Figures 15 and 16 are then computed by linearly interpolating the locations of the control points and the values of the radii of influence. We see that the result is a convincing modelling of the vortex motion. In a work in progress, we are now using implicit functions as a useful tool for the inverse problem.

7 Conclusion

The research presented in this paper is a first step towards a global study of complex structures undergoing deformations. We found implicit functions a very valuable tool to track and

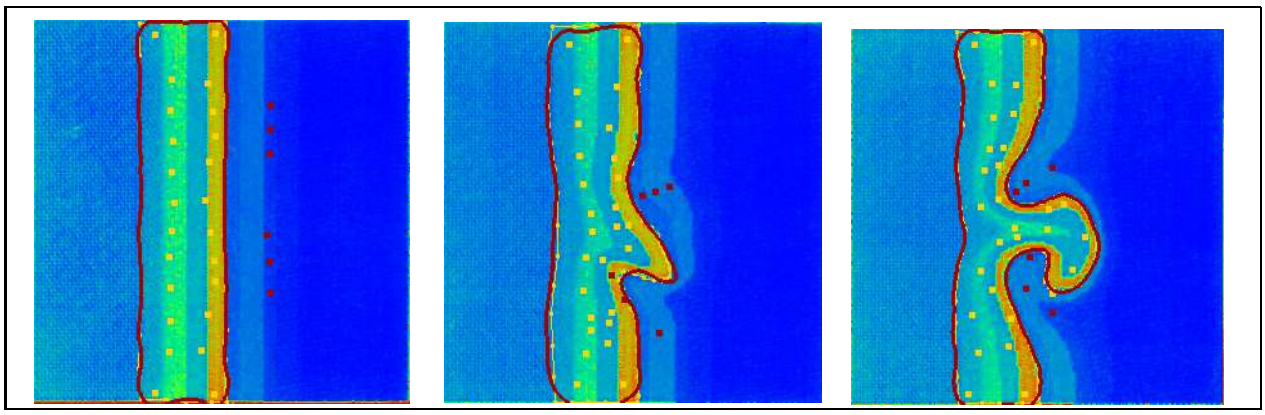

Figure 14: *Results.*

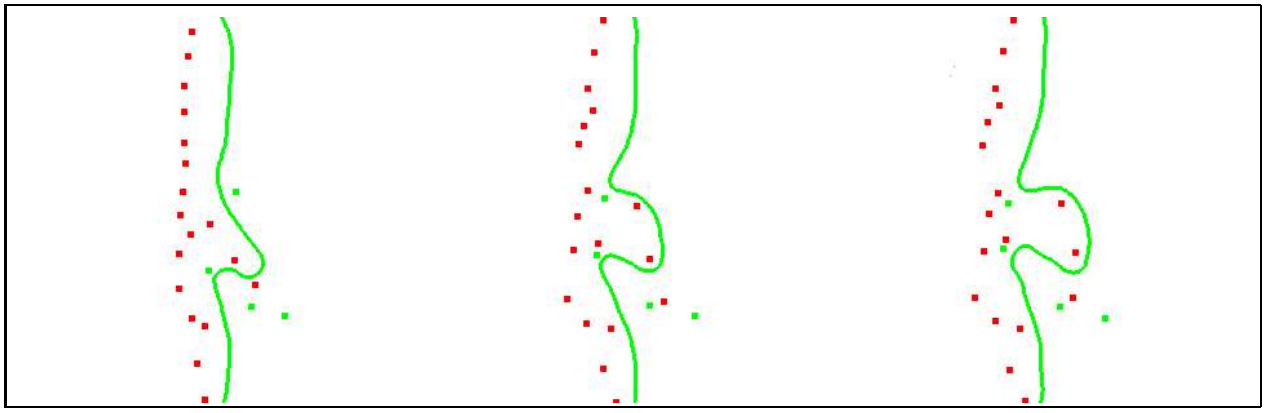

Figure 15: *Results. Positive control points are in red, negative in green. The green template is the resulting iso-contour corresponding to the control points and their radii of influence.*

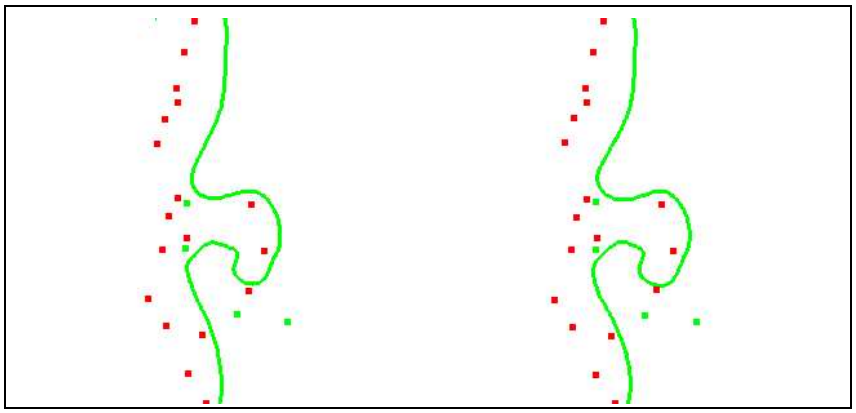

Figure 16: *Results.Positive control points are in red, negative in green. The green template is the resulting iso-contour corresponding to the control points and their radii of influence.*

model such deformations. For the motion of deformable structures, we are working on translating the physical motion of a structure into simpler kinematics on particles. We are also working on the modelling of even larger deformations, and we are using implicit functions to solve the inverse problem for the motion of deformable structures in natural images.

8 Appendix

In this appendix we give the computations of E's partial derivatives. These quantities are used in the conjugate gradient algorithm. From the definition of E, we see that we only need to compute the partial derivatives of local functions ϕ_j (see equation 4). One gets:

$$\frac{\partial \phi_j}{\partial x}(M) = 6a\frac{\|\overrightarrow{k_jM}\|^4}{{r_j}^6}(x - x_j) + 4b\frac{\|\overrightarrow{k_jM}\|^2}{{r_j}^4}(x - x_j) + \frac{2c}{{r_j}^2}(x - x_j) \tag{12}$$

and similarly for $\dfrac{\partial \phi_j}{\partial y}$. Also:

$$\frac{\partial^2 \phi_j}{\partial x^2}(M) = \frac{6a}{{r_j}^6}\ [\ \|\overrightarrow{k_jM}\|^4 + 4(x - x_j)^2\|\overrightarrow{k_jM}\|^2\ ]\ + \frac{4b}{{r_j}^4}\ [\ \|\overrightarrow{k_jM}\|^2 + 2(x - x_j)^2\ ]\ + \frac{2c}{{r_j}^2} \tag{13}$$

and similarly for $\dfrac{\partial^2 \phi_j}{\partial y^2}$.

$$\frac{\partial^2 \phi_j}{\partial x \partial y}(M) = \frac{12a}{{r_j}^6}(x - x_j)\ [2\|\overrightarrow{k_jM}\|^2(y - y_j)\ ] + \frac{8b}{{r_j}^4}(x - x_j)(y - y_j) \tag{14}$$

$$\frac{\partial^2 \phi_j}{\partial x_j \partial x}(M) = \frac{-6a}{{r_j}^6} [\ \|\overrightarrow{k_j M}\|^4 + 4(x-x_j)^2 \|\overrightarrow{k_j M} - \frac{4b}{{r_j}^4} [\ \|\overrightarrow{k_j M}\|^2 + 2(x-x_j)^2\ ] - \frac{2c}{{r_j}^2} \quad (15)$$

$$\frac{\partial^2 \phi_j}{\partial x_j \partial y}(M) = \frac{-24a}{{r_j}^6}(y-y_j)\ [\ \|\overrightarrow{k_j M}\|^2 (x-x_j)\ ] - \frac{8b}{{r_j}^4}(y-y_j)(x-x_j) \quad (16)$$

$$\frac{\partial^2 \phi_j}{\partial r_j \partial x}(M) = \frac{-36a}{{r_j}^7}(x-x_j)\|\overrightarrow{k_j M}\|^4 - \frac{16b}{{r_j}^5}(x-x_j)\|\overrightarrow{k_j M}\|^2 - \frac{4c}{{r_j}^3}(x-x_j) \quad (17)$$

$$\frac{\partial^3 \phi_j}{\partial x_j \partial x^2}(M) = \frac{-24a}{{r_j}^6} [\ \|\overrightarrow{k_j M}\|^2 (x-x_j) + 4(x-x_j)\|\overrightarrow{k_j M}\| + 2(x-x_j)^3\ ] - \frac{24b}{{r_j}^4}(x-x_j) \quad (18)$$

$$\frac{\partial^3 \phi_j}{\partial x_j \partial x \partial y}(M) = \frac{-24a}{{r_j}^6}(y-y_j)[\ \|\overrightarrow{k_j M}\|^2 + 2(x-x_j)^2\ ] - \frac{8b}{{r_j}^4}(y-y_j) \quad (19)$$

$$\frac{\partial^3 \phi_j}{\partial x_j \partial y^2}(M) = \frac{-24a}{{r_j}^6} [\ \|\overrightarrow{k_j M}\|^2 (x-x_j) + 2(y-y_j)^2 (x-x_j)\ ] - \frac{8b}{{r_j}^4}(x-x_j) \quad (20)$$

$$\frac{\partial^3 \phi_j}{\partial y_j \partial x^2}(M) = \frac{-24a}{{r_j}^6} [\ \|\overrightarrow{k_j M}\|^2 (y-y_j) + 2(y-y_j)(x-x_j)^2\ ] - \frac{8b}{{r_j}^4}(y-y_j) \quad (21)$$

$$\frac{\partial^3 \phi_j}{\partial y_j \partial x \partial y}(M) = \frac{-24a}{{r_j}^6}(x-x_j)[\ \|\overrightarrow{k_j M}\|^2 + 2(y-y_j)^2\ ] - \frac{8b}{{r_j}^4}(x-x_j) \quad (22)$$

Also:

$$\frac{\partial^3 \phi_j}{\partial r_j \partial x^2}(M) = \frac{-36a}{{r_j}^7}[\ \|\overrightarrow{k_j M}\|^4 + 4(x-x_j)^2 \|\overrightarrow{k_j M}\|^2\ ] - \frac{16b}{{r_j}^5}[\ \|\overrightarrow{k_j M}\|^2 + 2(x-x_j)^2\ ] - \frac{4c}{{r_j}^3} \quad (23)$$

and similarly for $\frac{\partial^3 \phi_j}{\partial r_j \partial y^2}$. Lastly:

$$\frac{\partial^3 \phi_j}{\partial r_j \partial x \partial y}(M) = \frac{-24a}{{r_j}^7}[\ 6\|\overrightarrow{k_j M}\|^2 (x-x_j)(y-y_j)\ ] - \frac{32b}{{r_j}^5}(x-x_j)(y-y_j) \quad (24)$$

References

[1] A. H Barr. Superquadrics and angle-preseving transforms. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[2] C. Blanc and C. Schlick. Extented field functions for soft objects. In *Implicit Surfaces'95*, pages 21–32, Grenoble, France, April 1995.

[3] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, pages 235–256, July 1982.

[4] J. Bloomenthal. Polygonalization of implicit surfaces. *Computer Aided Geometric Design*, 5:341–355, 1988.

[5] James H. Clark. Parametric curves, surfaces and volumes in computer graphics and computer-aided geometric design. Technical Report 221, Stanford University, November 1981.

[6] I. Cohen and L. D. Cohen. A hybrid hyperquadric model for 2-D and 3-D data fitting. Technical Report 2188, Institut National de Recherche en Informatique et Automatique, February 1994.

[7] T. F. Cootes, C. J. Taylor, D. H. Cooper, and Graham J. Active shape models–their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[8] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, pages 167–187, 1987.

[9] M. P. Gascuel. Modelisation surfacique d'objets: Partie 2: Fonctions implicites. *Courses Notes (in French)*, pages 1–22, 1992.

[10] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, New-York and London, 1981.

[11] I.L. Herlin, C. Nguyen, and Ch. Graffigne. Stochastic segmentation of ultrasound images. In *ICPR Proceedings*, The Hague, Holland, August 30–September 3 1992.

[12] M. J. Laszlo. Fast generation and display of iso-surface wireframes. *CVGIP: Graphical Models and Image Processing*, 54(6):473–483, 1992.

[13] J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. *ACM Siggraph Proceedings*, pages 27–34, July 1993.

[14] S. Muraki. Volumetric shape description of range data using blobby model. *Computer Graphics*, 25(4):227–235, July 1991.

[15] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Objects modeling by distribution function and a method of image generation (in japanese). *Trans. IEICE*, J68-D(4):718–725, 1985.

[16] A. Pentland and K. Mase. Automatic visual recognition of spoken words. Technical Report 117, M.I.T. Media Lab Vision Science, August 1989.

[17] C.H. Reinsch. Smoothing by spline functions I. *Numerische Mathematik*, 10:177–183, 1967.

[18] S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. *Computer Graphics*, 25(4):247–250, July 1991.

[19] T. Taxt, A. Lundervold, and B. Angelsen. Noise reduction and segmentation in time-varying ultrasound images. In *10th International Conference on Pattern Recognition*, Atlantic City, New Jersey, USA, June 1990.

[20] H. Tek and B. B. Kimia. Shock-based reaction-diffusion bubbles for image segmentation. Technical report, Brown University, August 1994.

[21] J. A. Thorpe. *Elementary Topics In Differential Geometry*. Springer-Verlag, New-York Heidelberg Berlin, 1979.

[22] B. Wyvill and G. Wyvill. Field functions for implicit surfaces. *The Visual Computer*, 5:75–82, December 1989.

[23] G. Wyvill, C. McPheeters, and B. Wyvill. Animating soft objects. *The Visual Computer*, 2:235–242, August 1986.

[24] G. Wyvill, C. McPheeters, and Brian Wyvill. Data structures for soft objects. *The Visual Computer*, 2:227–234, August 1986.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur

INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)

ISSN 0249-6399